IEEE P802.15
Wireless Personal Area Networks

NOTE: To change <title> and other required fields, select File→Properties and update the appropriate fields in the Summary tab. DO NOT replace field codes with text. After updates are entered, delete this paragraph and update all fields (ctl-A then F9) Note: dates will not be updated until document is saved. After fields are updated, delete this paragraph..

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)		
Title	**Non-Collaborative MAC Mechanisms**		
Date Submitted	25 June 2001		
Source	Nada Golmie NIST 100 Bureau Dr. Stop 8920 Gaithersburg MD 20899	Voice: Fax: E-mail:	(301) 975-4190 (301) 590-0932 nada@nist.gov
	Jie Liang Texas Instruments 12500 TI Boulevard, MS8649 Dallas, TX 75243	Voice: E-mail:	(214)480-4105 liang@ti.com
	The is proposed text for clause 14.2 of the Recommended Practice Document		
Abstract	[]		
Purpose	[]		
Notice	This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.		
Release	The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15.		

1. Adaptive Packet Selection and Scheduling

1.1 Introduction

In this section, we introduce several methods that enhance the performance of Bluetooth and 802.11 networks through the use of adaptive packet selection and scheduling for the Bluetooth devices. These methods do not require the collaboration between the 802.11 devices and the Bluetooth devices. Therefore, they belong to the general category of non-collaborative coexistence mechanisms. Furthermore, these mechanisms, which adapt the packet types and transmission timing to the channel condition of the current hop, can be implemented mostly through MAC layer enhancement without significantly changing the hardware structure of most Bluetooth implementations.

The key idea for adaptive packet selection and scheduling methods is to adapt the transmission according to channel conditions. For instance, if the channel is dominated by interference from 802.11b network, packet loss will be mainly due to collisions between BT and 802.11 systems, instead of bit errors resulting from noise. Packet types that do not include FEC protection could provide better throughput if combined with intelligent packet scheduling. The foundation for the effectiveness of these types of methods is to be able to figure out the current channel conditions accurately and timely. Channel estimation can be done in a variety of ways: RSSI, HEC decoding profile, BER and PER profile, and an intelligent combination of all of the above.

Editorial Note (Jie Liang): we need a section talking exclusively about estimating channel conditions and it can be shared between AFH chapter and packet selection and scheduling chapter.

1.2 Adaptive Packet Selection

- BT packet types for SCO and ACL
- Methods of adaptive packet selection

1.3 Packet Scheduling for SCO Links

1.3.1 BT SCO Link

1.3.2 SCO Scheduling algorithm for coexistence enhancement

1.3.3 Performance simulation

1.3.4 Summary

1.4 Packet Scheduling for ACL Links

In this section, we describe packet scheduling techniques that can be used to alleviate the impact of interference. We devise a mechanism for the Bluetooth MAC scheduler consisting of two components:

1. Interference Estimation
2. Master Delay Policy

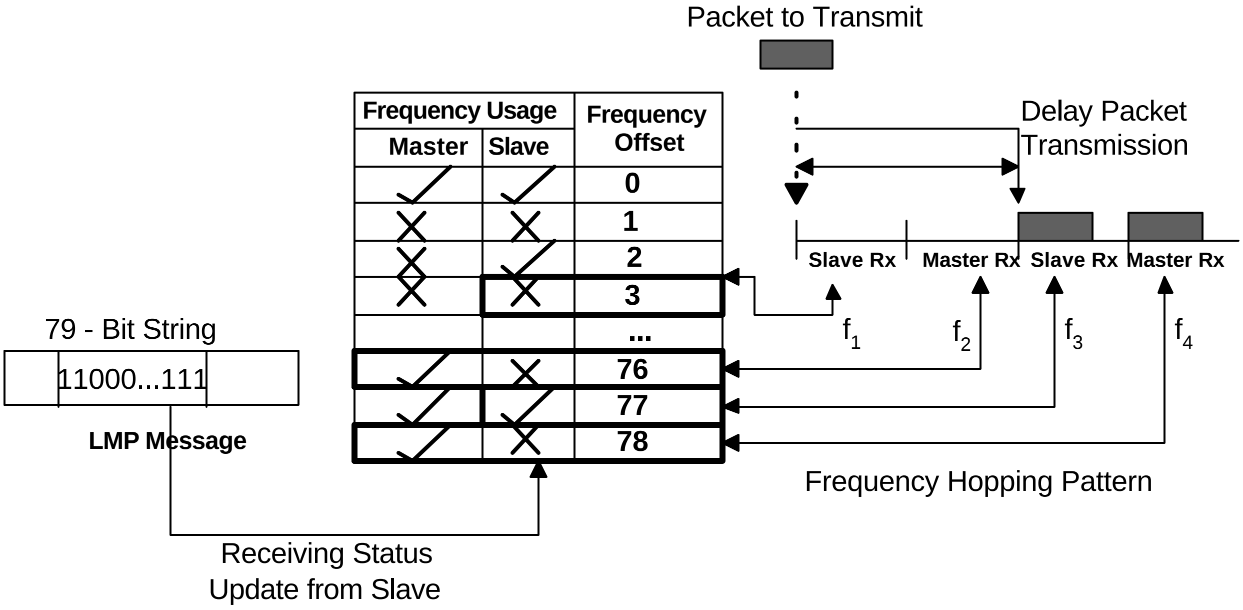
In the *Interference Estimation* phase, the Bluetooth device detects the presence of an interfering device occupying a number of frequencies in the band. In this sequel, interfering devices are assumed to be WLAN DSSS systems.
In order to detect the presence of interference, the Bluetooth device maintains a *Frequency Usage Table* where a bit error rate measurement, **BER_f**, is associated to each frequency as shown in Figure 1. Note that, a frame error rate or a packet loss measure can be used instead of the BER. Frequencies are classified according to a criteria that measures the level of interference in the channel and marked *used* or *unused* depending on whether their corresponding BER is above or below a threshold value, **BER^T**, respectively.
This *Frequency Usage Table* is maintained at each receiver's side for both master and slave devices.

Use	Frequency Offset	BER$_f$
✓	0	10^{-3}
✗	1	10^{-1}
✗	2	10^{-2}
✗	3	10^{-1}
	...	
✓	76	10^{-4}
✓	77	10^{-3}
✓	78	10^{-3}

Figure 1: Frequency Usage Table

The *Master Delay Policy* makes use of the measurements collected during the *Interference Estimation* phase in order to avoid a packet transmission in a "bad" receiving channel, or a channel with a high level of interference. The basic idea is to "wait" for or choose an *unused* frequency for the receiver in the frequency hopping pattern. Thus the transmitter needs to consult the receiver's *Frequency Usage Table* before transmitting any packets. Alternatively, the receiver, can send status updates on its usage table to the transmitter.

In Bluetooth, since the master device controls all transmissions in the piconet, the delay rule has to be implemented only in the master device. Furthermore, since following each master's transmission, there is a slave transmission, the master checks both the slave's receiving frequency and its own receiving frequency before chosing to transmit a packet in a given frequency hop as illustrated in Figure 2.



Frequency Usage		Frequency Offset
Master	Slave	
✓	✓	0
✗	✗	1
✗	✓	2
✗	✗	3
		...
✓	✗	76
✓	✓	77
✓	✗	78

Figure 2: Delay Scheduling Policy at Bluetooth Master

The main steps of the scheduling policy are summarized as follows.

- Slave's End.
  1. For every packet received, update **BER_f** which is an average value of the BER per frequency.
  2. Every update interval, **U**, refresh the *Frequency Usage Table* by marking the frequencies, and
  3. Send a status update message to the Master;
- Master's End.
  1. For every packet received, update **BER_f** which is an average value of the BER per frequency.
  2. Every update interval, **U**, refresh the *Frequency Usage Table*, and
  3. Before sending a packet, check slave's receiving frequency and master's following receiving frequency, delay transmission until both master and slave's receiving frequencies are available.

1.4.1 Implementation Considerations

One of the advantages in using this scheduling policy is that it does not require any changes in the FCC rules. In fact, title 47, part 15 of the FCC rules on radio frequency devices, allows a frequency hopping system to recognize the presence of other users within the same spectrum band so that it adapts its hopsets to avoid hopping on occupied channels. However, coordination among hopping frequency systems in order to avoid simultaneous channel occupancy is not allowed.

Furthermore, scheduling in the Bluetooth specifications is vendor implementation specific. Therefore, one can easily implement a scheduling policy with the currently available Bluetooth chip set. Most importantly, the proposed scheduling algorithm does not require any changes to the Bluetooth frequency hopping pattern which is implemented in ASICs, and devices implementing scheduling can easily interoperate with other devices that do not.

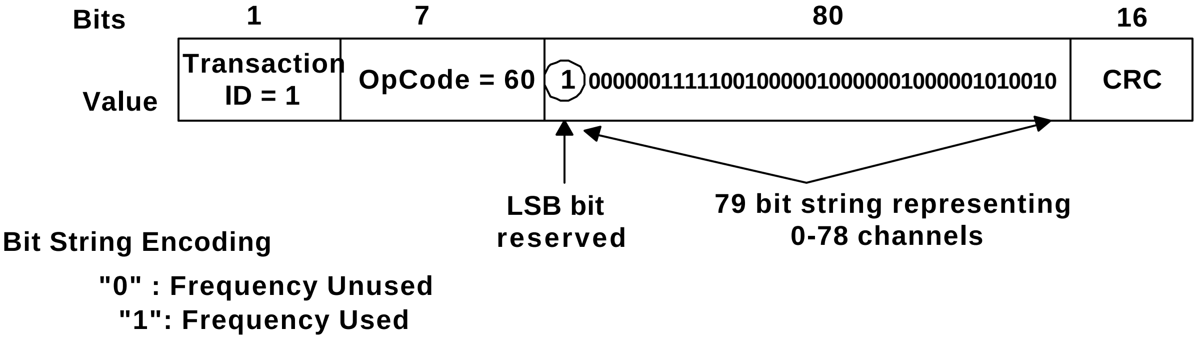


Figure 3: LMP Interference Status PDU

As far as the status update message is concerned, we define an **LMP_Interference_Status** PDU as shown in Figure 3. We use an **Op_code** value of **60** and set the **Transition ID** to **1** in order to indicate that the message is sent from the slave to the master. The content field uses 10 bytes to encode the slave's *Frequency Usage Table*. In fact, we reserve one bit for future use, and map the 79 channels in the *Frequency Usage Table* to a 79-bit string of 0's and 1's indicating the *used* and unused receiving frequencies respectively.

1.4.2 Numerical Results

We simulate our proposed scheduling policy. We use a 4-node topology consisting of two Bluetooth nodes (1 master and 1 slave) and two WLAN devices (1 Access Point and 1 Mobile device). The Bluetooth devices are located at (0,0) meters for the slave device and (1,0) meters for the master device. The WLAN devices are located at (0,15) meters for the AP and (0,d) for the mobile device. We assume that WLAN devices implement the IEEE 802.11b specifications at 11 Mbits/s. The WLAN mobile is assumed to be transmitting data to the AP which responds with ACK messages. The WLAN offered load is assumed to be 50% of the channel capacity, the data packet size is set to 8000 bits (including the MAC header) and the packet interearrival time is assumed to be exponential with a mean equal to 1.86 ms.

We use three types of Bluetooth packet encapsulations, namely, DM1, DM3, and DM5, that occupy 1, 3 and 5 slots respectively. The offered load for Bluetooth is set to 30% of the channel capacity which corresponds to a packet interarrival of 2.91 ms, 8.75 ms and 14.58 ms for DM1, DM3 and DM5 packets respectively.

The transmitted power for Bluetooth and WLAN is fixed at 1mW and 25 mW respectively.

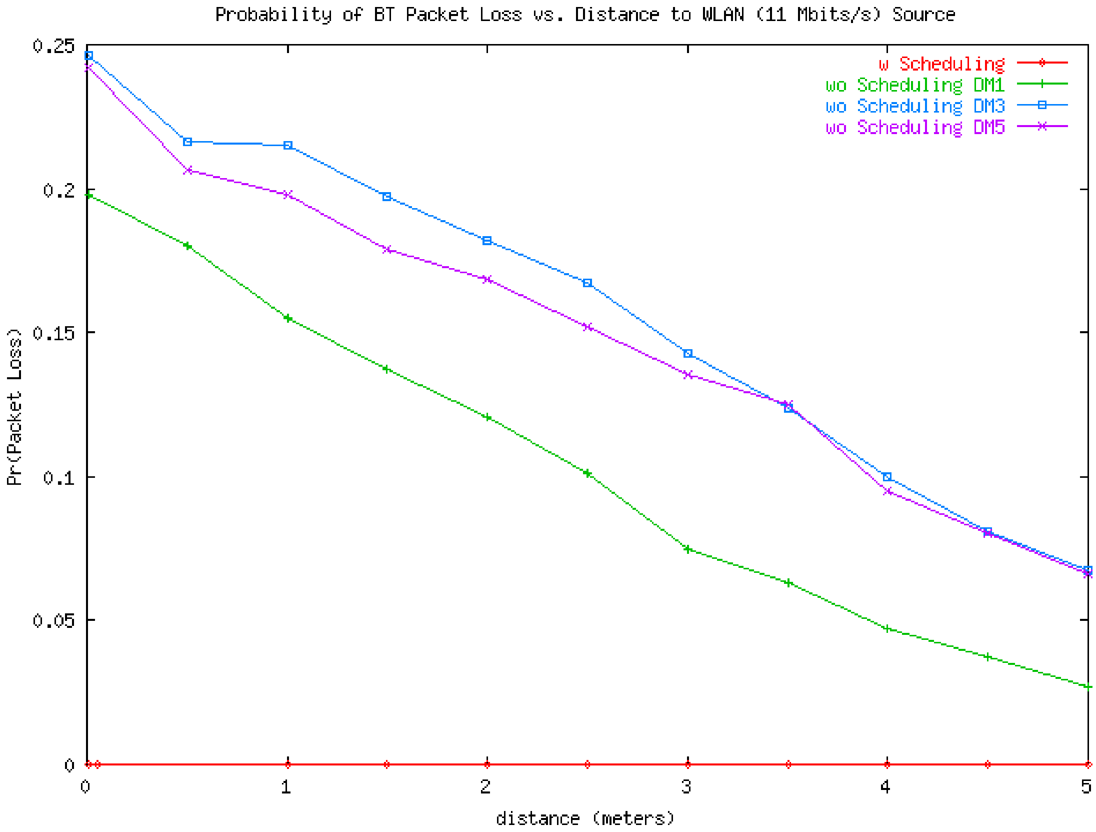


Figure 4: Effect of Scheduling on Bluetooth - Probability of Packet Loss

Figures 4 and 5 give the packet loss and the mean access delay respectively measured at the Bluetooth slave for varying distances of the interference source from the Bluetooth receiver.

From Figure 4 we observe that using the scheduling policy, leads to a packet loss of zero. We are basically able to avoid the channels occupied by the interfering system. When no scheduling policy is used the packet loss is ~ 24% for DM5, and DM3, and 19% for and DM1 packets respectively when the Bluetooth receiver is at a distance of 0.005 meters from the interference source. As the distance from the interference source is increased the packet loss drops to around 2.7% for DM1 packets. It is still around 6.7% for DM3 and DM5 packets.

For DM1, we observe an increase in delay from 1.6ms to 2.6ms when the scheduling policy is applied. On average the scheduling policy yields to a delay increase of 1ms (~1.6 Bluetooth slots). On the other hand, the scheduling policy reduces the delays by 0.8 ms and 2.6 ms for DM3 and DM5 respectively. Thus, delaying transmission to avoid bad channels pays off for packets occupying more than one slot. Note that, when bad channels are used, packets are dropped and have to be retransmitted which yields large delays. This effect does not apply to DM1 packets since they occupy only one slot.

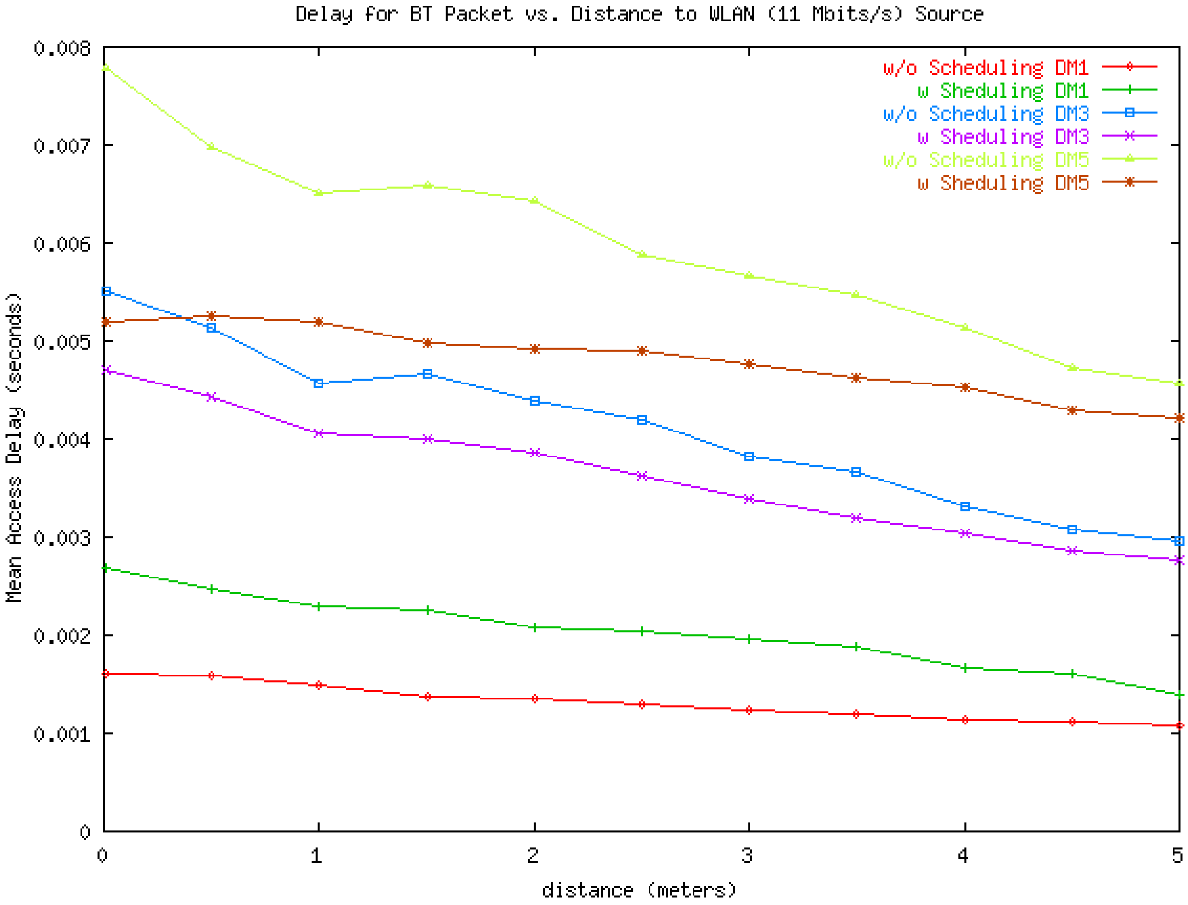


Figure 5: Effect of Scheduling on Bluetooth - Mean Access Delay

In summary, we note that the scheduling policy is effective in reducing packet loss and delay (especially for multi-slot Bluetooth packets). Another advantage worth mentioning, are the additional savings in the transmitted power since packets are not transmitted when the channel is bad. Moreover, we note that by avoiding channels occupied by other devices, we eliminate interference on the other system sharing the same spectrum band. Figure ~\ref{wlan-pk} shows the packet loss for the WLAN Mobile device (receiving ACKs). We note that scheduling reduces the ACK packet loss to zero. Therefore scheduling can be considered as a neighbor friendly policy. Note that the packet loss at the WLAN AP located at (0,15) m is negligible in this case since the Bluetooth signal is too weak.

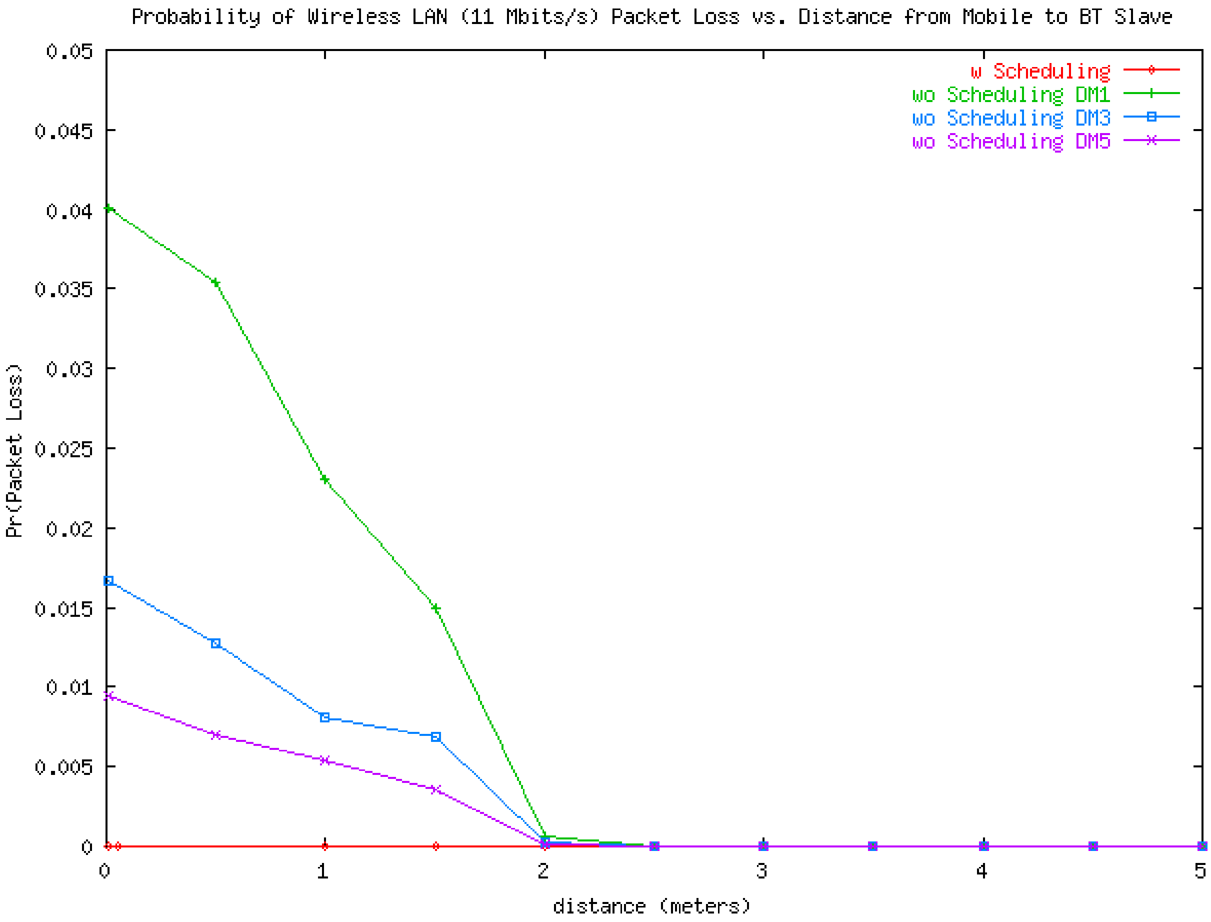


Figure 6: Impact of MAC Scheduling on the WLAN Mobile Device

Finally, we note that scheduling policy proposed here works only with data traffic since voice packets need to be sent at fixed intervals. However, if the delay variance is constant and the delay can be limited to a slot (as was shown here), it may be worthwhile to use DM packets for voice using the same scheduling technique proposed here. This will constitute the basis of future work.